Natural Language Processing Coursework

Link to Colab: Colab Notebook

Wenjia Wang (ww2321)
ww2321@ic.ac.uk

Jiaqi Zhao (jz5421)
jz5421@ic.ac.uk

Yizhou Wu (yw7421)
yw7421@ic.ac.uk

Abstract

This report outlines our design process and results for SemEval 2022 Task 4 Subtask 1 of identifying whether patronizing and condescending language (PCL) is present in a given text. We developed a binary classification model employing the **Bidirectional Encoder Representations from Transformers (BERT(1))** that outperforms the RoBERTa-base baseline. The final submission result of our model is 0.6354 for precision and 0.5552 recall MAE and 0.5926 for F1.

1 Introduction

Condescending language and condescending language (PCL) can cause the end of conversations and the division of communities. The detection of PCL is a critical problem in the field of linguistics. Our datasets have three parts: a training set for building models, a validation set for model selection and hyperparameter tuning, and a test set. In the following report, the validation set will be referred to as the official development set.

2 Data Analysis of the training data

The 'Don't Patronize Me!'(DPM) training set contains 10,637 paragraphs extracted from news stories, which have been annotated to indicate the presence of PCL at the text span level.

2.1 Analysis of the class labels

The following tables show the proportion of class labels and how they correlate with the features of the data, including the input length and keywords.

This training dataset is highly imbalanced according to the Table 1, most data are classified as non PCL. Table 2 shows the proportion in two labels for different keywords.

Label	0	1	2	3	4
No. paragraph	8528	947	144	458	391
PCL	False	False	True	Ture	Ture

Table 1: Training dataset distribution under the Label

	Label 0	Label 1
migrant	0.956801	0.043199
in-need	0.84658	0.15342
vulnerable	0.926852	0.073148
homeless	0.84494	0.15506
women	0.933645	0.066355
refugee	0.919476	0.080524
immigrant	0.961357	0.038643
disabled	0.919261	0.080739
hopeless	0.884577	0.115423
poor-families	0.850385	0.149615

Table 2: Proportion of Label with different keywords

	Label 0	Label 1
<=20	0.913921	0.086079
21-40	0.919318	0.080682
41-60	0.903421	0.096579
61-100	0.889442	0.110558
>100	0.855311	0.144689

Table 3: Proportion of Label under input length

Table 3 represents how different input length correlates with the classification result. With an increasing input length, the sentences seems to be less patronizing and condescending.

2.2 Qualitative Assessment of the Dataset

The performance of most machine learning models tends to be poor since the class distribution is imbalanced. Modifications are required to avoid simply predicting the dominant class in all cases. In addition, metrics like accuracy can't stand for the robustness of the model. Alternate method like F1 Score for evaluating predictions on imbalanced examples is required. We aim to increase minor class identification instead of overall accuracy.

For example, since most of the sentences with the keyword "women" are non-PCL, the algorithm tends to classify all data with it to label 0, which will significantly harm the model performance on the minority class. Additionally, the dataset itself might be subjective. The sentences may be classified manually by people with different cultures, gender and age demographics, who may have controversial views on the same sentence.

3 Modelling

3.1 Data Processing

We made our model case-insensitive because the case has no bearing on PCL's decision. We kept the entire feature space without punctuation, stemming, lemmatisation and stop words since we discovered these text pre-processing techniques can occasionally alter the meaning of the original data, which is critical to the PCL of the sentence. The following is the processing we did,

- **Drop the null paragraphs**
  10,468 paragraphs left after deletion.

- **Binary Classification**
  The paragraphs labelled 0, 1 are considered 0, while others are considered 1. A new column "labels" stores this information.

- **Feature Combination**
  We combined all features into one new feature for all datasets.

- **Tokenization**
  The BertTokenizer was used to tokenize the words in dataset. To create the BERT word embedding form, a [CLS] token is inserted at the beginning of the first sentence and a [SEP] token is put at the end of each sentence.

3.2 Creative Improvement

The bias in the imbalanced training data set can affect many machine learning models, resulting in some ignoring the minority class entirely. This is a problem as it's usually the minority class on which predictions are more significant. One approach for dealing with class imbalance is to randomly resample the training data set. This approach is oversampling, and it's a basic way to rebalance the class distribution in an imbalanced dataset.

Therefore, we utilized the method of oversampling, duplicating examples from the minority class and adding them to the training data set.

3.3 Hyperparameter Tuning

For hyperparameter tuning, we randomly sampled a portion of training set. We chose a small batch size 4 since training consumes lots of memory. 2, 4, 5, 10 epochs were tested since large epoch number will cause overfitting. Finally, 4 epochs was selected with the best validation performance.

We used the Linear Warmup With Cosine Annealing to adjust the learning rate of model. It is a learning rate schedule in which the learning rate is increased linearly for updates and then annealed according to a cosine schedule. We set the learning rate from $1e^{-6}$ to $1e^{-3}$. From Table 4, $1e^{-6}$ is the best learning rate for training.

	1e-6	1e-5	1e-4	1e-3
Epoch 1	0.858	0.823	0.468	0.124
Epoch 2	0.843	0.794	0.468	0.468
Epoch 3	0.800	0.794	0.468	0.107
Epoch 4	0.822	0.803	0.307	0.107
Epoch 5	0.823	0.818	0.468	0.106

Table 4: F1 Score with different learning rate

3.4 Model Configuration

- learning rate scheduler type: cosine
- learning rate: $1e^{-6}$
- Epoch number: 4
- Batch size: 4
- Warmup steps: 200

3.5 Model Performance

After our analysis of some unimportant categories (detailed in part 4 of the report: Analysis), our final model does not include the category of ID and ranks 4^{th} by precision and 7^{th} by F1 Score on codalab for task 1. The evaluation metrics as follows,

	Precision	Recall	F1
Test Set	0.635379	0.555205	0.592593

Table 5: Evaluation Metrics of the test dataset

4 Analysis

1) To what extent is the model better at predicting examples with a higher level of patronising content? Justify your answer.

We divided the five categories into two, but in fact, 0 to 4 all show the existence of PCL in varying degrees. Putting 0, 1 into a new category and 2, 3, 4 into a new category may not be the best option. We can assign a weight to each label.

2) How does the length of the input sequence impact the model performance?

	F1	Number of paragraph
< 20	0.691526	100
20 - 40	0.779022	400
40 - 60	0.733511	299
60 - 100	0.738128	193
> 100	0.676297	55

Table 6: F1 score of different length of input

We used 10% of the training set as an internal dev set to verify input length. The input length $>$ 100 has the lowest F1 score, followed by input length $<$ 20 (see Table 6). We concluded that the F1 score roughly increases with increasing length, then decreases with increasing length after reaching a peak. So, when the input is too short or too long, our model does not perform well. The accuracy is relatively high when the input length is moderate.

Possible reasons are the model may capture less key information in short paragraphs, it's more likely to make errors when there's a confusing word(e.g., multiple semantics) and context can't be determined by other words. For overly long sentences, there's a greater chance more confusing words will affect the prediction. So, input sequences of moderate length are best predicted.

3) To what extent does the categorical data provided influence the model predictions.

	With ID	No ID	No ID,Country
Epoch 1	0.468085	0.265913	0.539334
Epoch 2	0.368085	0.520045	0.330357
Epoch 3	0.627976	0.558212	0.681077
Epoch 4	0.675114	0.762873	0.670251
Epoch 5	0.691526	0.764706	0.702857
Epoch 6	0.795710	0.740091	0.707602
Epoch 7	0.751861	0.840000	0.709302
Epoch 8	0.726710	0.866986	0.795710

Table 7: F1 Score with different categories

From our perspective, the **ID** feature is unnecessary to consider. From Table 7, the overall performance of the model without ID is much better than with it. This may be the redundant information influencing the model accuracy.

The **Keyword** may influence whether the current paragraph contains PCL or not, but cannot determine PCL. In Table 2, ”hopeless” can either be in PCL or non-PCL. However, because ”women” is only a proper noun, it may have less influence on the classification result than other keywords.

We consider the feature of **Country** may affect the prediction of the model. The countries with political turmoil or wars may be with more texts containing PCL. Table 7 confirms that a model with country information have a better F1 score.

5 Conclusion and Future Work

We achieve satisfying results in this task. Nevertheless, we evaluate the model's performance by early stopping the running epoch with a smaller data set. This to some extent alleviates the occurrence of overfitting. One potential way to improve the model performance can be adding weights to different categories such as keywords.

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. cite arxiv:1810.04805. Comment: 13 pages.

[2] C. Pérez-Almendros, L. Espinosa-Anke, S. Schockaert. Don't Patronize Me! An Annotated Dataset with Patronizing and Condescending Language towards Vulnerable Communities, 2020. cite arXiv:2011.08320. Comment: 12 pages.